c) Amendments to the Claims

Please cancel claims 3 and 15, without prejudice.

Please amend the remaining claims as follows:

1. (Currently amended) A method including:

in a queue, writing a first instruction of a plurality of instructions to a first location indicated by a write pointer; the plurality of instructions being written to the queue as a set of a predetermined number of instructions, and the first instruction of the plurality of instructions being indicated as invalid on account of being outside a trace of instructions;

making a qualitative determination whether or not to retain the first instruction within the queue; based on the indicated invalidity of the first instruction;

if the qualitative determination is to retain the first instructions, then advancing the write pointer to indicate a second location within the queue into which to write a second instruction; and

if the qualitative determination is not to retain the first instruction, then maintaining the write pointer to indicate the first location within the queue into which to write the second instruction, so that the first instruction is overwritten by the second instruction.

2. (Original) The method of claim 1 wherein the qualitative determination includes examining a valid bit associated with the first instruction to determine validity of the first instruction, making the qualitative determination to retain the first instruction if the valid bit indicates the first instruction is being valid, and making the qualitative determination not to retain the first instruction if the valid bit indicates the first instruction as being invalid.

3. (Canceled)

4. (Original) The method of claim 2 wherein a plurality of instructions are written to the queue in a set of a predetermined number of instructions, and wherein at least one instruction of the set is indicated as being invalid on account of a branch misprediction relating to a branch instruction upstream of the at least one instruction in a stream of instructions.

5. (Original) The method of claim 1 wherein the first instruction comprises a first microinstruction.

6. (Original) The method of claim 5 wherein the first microinstruction is written to the queue from a microinstruction cache.

7. (Original) The method of claim 6 wherein the first microinstruction is part of a trace of microinstructions received from the microinstruction cache.

8. (Original) The method of claim 6 wherein the first instruction is received from an instruction source operating in a first clocking domain into the queue and read from the queue to an instruction destination operating in a second clocking domain.

9. (Original) The method of claim 1 wherein the first instruction is received into the queue as part of a set of instructions comprising a first predetermined number of instructions and read from the queue to an instruction destination as part of a second set of instructions comprising a second number of instructions.

10. (Original) The method of claim 1 wherein the first instruction is written from a source to a destination, and wherein the queue comprises a first path between source and destination, the method including propagating the first instruction from the source to the destination via a second path, not including the queue, if the queue is empty.

11. (Original) The method of claim 10 including selecting between the first and second paths to receive the first instruction for propagation to the destination.

12. (Original) The method of claim 1 wherein the queue includes a first portion to support a first thread within a multithreaded environment and a second portion to support a second thread within the multithreaded environment, and wherein the first location into which the first instruction is written is located in the first portion if the first instruction comprises part of the first thread.

13. (Currently amended) Apparatus comprising:

a queue ~~for buffering~~ to buffer a first instruction propagated from a source to a destination; and

write logic to make a qualitative determination whether or not to retain the first instruction within the queue; if the qualitative determination is to retain the first instruction, to advance a write pointer to indicate a second location within the queue into which to write a second instruction; and, if the qualitative determination is not to ~~return~~ retain the first instruction, to maintain the write pointer to indicate the first location within the queue into which to write the second instruction, so that the first instruction is overwritten by the second instruction~~.~~;

wherein the first instruction is written to the queue as part of a set including a predetermined number of instructions, and wherein at least one instruction of the set is indicated as being invalid on account of being outside a trace of instructions.

14. (Original) The apparatus of claim 13 wherein the write logic is to examine a valid bit associated with the first instruction to determine validity of the first instruction, to make the qualitative determination to retain the first instruction if the valid bit indicates the first instruction as being valid, and to make the qualitative determination not to retain the first instruction if the valid bit indicates the first bit as being invalid.

15. (Canceled)

16. (Original) The apparatus of claim 14 wherein the first instruction is written to

the queue as part of a set of a predetermined number of instructions, and wherein at least one instruction of the set is indicated as being invalid on account of a branch misprediction relating to a branch instruction upstream of the at least one instruction in a stream of instructions.

17. (Original) The apparatus of claim 1 wherein the first instruction is a first microinstruction.

18. (Original) The apparatus of claim 17 wherein the source from which the first microinstruction is written to the queue comprises a microinstruction cache.

19. (Original) The apparatus of claim 18 wherein the first microinstruction is part of a trace of microinstructions received from the microinstruction cache.

20. (Original) The apparatus of claim 13 wherein the queue comprises a first path between the source and the destination, the apparatus including a second path between the source and destination, not including the queue, and wherein the write logic directs the first microinstruction to be propagated between the source and destination via the second path if the queue is empty.

21. (Currently amended) A machine-readable medium storing a sequence of instructions that, when executed by machine, cause the machine to perform the steps of:

in a queue, writing a first instruction of a plurality of instructions to a first location indicated by a write pointer; the plurality of instructions being written to the queue as a set of a predetermined number of instructions, and the first instruction of the plurality of instructions being indicated as invalid on account of being outside a trace of instructions;

making a qualitative determination whether or not to retain the first instruction within the queue; based on the indicated invalidity of the first instruction;

if the qualitative determination is to retain the first instructions, then advancing the write pointer to indicate a second location within the queue into which to write a

second instruction; and

A2
B1

if the qualitative determination is not to retain the first instruction, then maintaining the write pointer to indicate the first location within the queue into which to write the second instruction, so that the first instruction is overwritten by the second instruction.

22. (Original) The machine-readable medium of claim 21 wherein the sequence of instructions cause a multiprocessor to perform the step of examining a valid bit associated with the instruction to determine validity of the first instruction, to make the qualitative determination to retain the first instruction if the valid bit indicates the first instruction as being valid, and to make the qualitative determination not to retain the first instruction if the valid bit indicates the first instruction as being invalid.